In the last chapter we established the fact that the computer has a memory made up of many boxes and these boxes are each named or labeled with an address. Furthermore the boxes are used to hold those mysterious instructions that will enable the computer to do our bidding by following a program that we have carefully set down for it. In this chapter we're going to spend some more time on the memory and find out exactly how to control what gets into it.

Memory Makeup. There are actually two programs that we are concerned with at any given time. One, quite naturally is our own that we have written and expect the computer to execute for us. The second is the monitor program, a phantom that is always in the background and makes it possible for us to load and execute our programs, read the keyboard, drive the displays, operate the tape recorder interface, and so on. It is a fact that the type of memory circuits used in a computer capable of storing programs of our own writing, will also lose those programs the second the power is turned off. We can see this by examining the memory, switching off the power for a moment, switching it back on, and then reexamining the same memory locations.

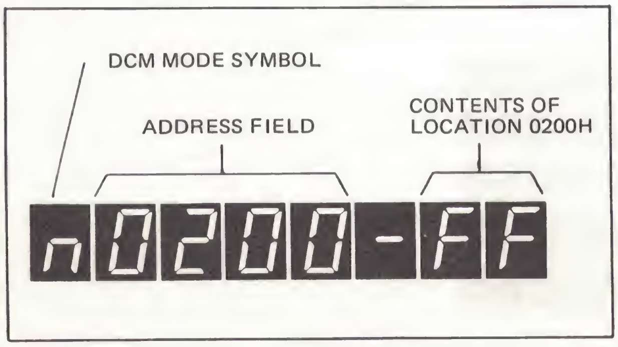

Fig. 2-1. In the DCM mode the computer will display the DCM symbol, a 4-digit address and a two digit "content." Thus in one glance we can see which memory location is being examined and what the contents of that location are.

Try it, writing the contents of each memory location so you can recheck after turning the power off and on.

Enter:	See Displayed:	Write Contents:
DCM 0 2 0 0 NXT	n0200-	
NXT	n0201-	
NXT	n0202-	
NXT	n0203-	
NXT	n0204-	

Now turn the computer off for a few seconds and then turn it back on. Check the same locations and write the contents of each location in the space provided below.

Enter:	See Displayed:	Write Contents:
DCM 0 2 0 0 NXT	n0200-	
NXT	n0201-	
NXT	n0202-	
NXT	n0203-	
NXT	n0204-	

If you go back now and check the same locations against the contents that you recorded earlier you will undoubtedly notice that the contents of the memory locations have changed. This is a characteristic of these memory integrated circuits called RAMs. They can have data written into them and they will store it intact until new data replaces old, but they will lose that data the second the power is removed. When power is reapplied the RAM circuits will come back up with completely random data contained in them. In many programs this will require that the program initialize the memory locations with some known data before they can be used. But we'll cover that in good time.

The ia7301 has 1024 RAM Read/Write memory locations. We can address those locations by using addresses in the range from 0000H to 03FFH. With the exception of a few RAM locations that are used by the computer as it executes the monitor program, most of those 1024 locations are available for your use to hold programs and data. There's a lot of talk about how much memory various computers have, but as you will discover, 1024 RAM locations is plenty for some very complex programs. By the time this course is completed you should be have a very good idea how much memory will be required for those little pet projects you've been wanting to do with your computer. At that time, you'll want to check out the various Iasis boards and software products designed to extend the capacity of the ia7301; it'll go as far as you want.

If the RAM memory is used for holding your programs, where does the monitor program reside? The answer to that is tied into the whole problem of memory volatility. Volatility is an adjective applied to those memory circuits that lose their data when power is removed for a second. We have already seen that RAMs are volatile; turn the power off for the evening and your program will have gone where all good programs go. The next morning you'll have to reload it if your're going to execute that favorite checkbook routine. Well, there are some memory circuits that are nonvolatile. You can turn the power off as many times as you want and every time you turn it back on, your program will be there ready to go. Most nonvolatile memories fall into the category of ROMs (Read Only Memories). These memories from which the computer can read instructions and hence, programs, but will not accept new data. How do the instructions find their way into the ROMs? They might be put there according to your written instructions by the semiconductor manufacturer that makes the ROM. Or, they may be put in electrically by a special piece of equipment called a Programmer. The memories are then called PROMs (Programmable Read Only Memories). Sometimes these PROMs can be erased by using strong ultraviolet light or special erasing voltages and then reprogrammed with a new set of instructions. They are then called EROMs (Erasable Read Only Memories). Either PROMs or EROMs are suitable for holding the instructions that make up the monitor program, since both types retain data when power is removed. The monitor of the ia7301 requires 1024 locations to hold the instructions that make up the programs beginning at location 8000H and ending at 83FFH. Let's use the ia7301 to experiment with the two basic types of memory.

Writing Into Memory. Using DCM try to access memory location 0200H. Write the contents of that location in the space provided.

Enter:	See Displayed:	Write Contents:
DCM 0 2 0 0 NXT	n0200-	______________
F F	n0200-FF	______________
NXT	n0201-	
DCM 0 2 0 0 NXT	n0200-FF	______________

We have successfully written FFH into the memory location 0200H. That location now contains

FFH and will until we choose to write a different two digit data word into it.

Enter:	See Displayed:	Write Contents:
DCM 0 2 0 0 NXT	n0200-	
0 0	n0200-00	
NXT	n0201-	
DCM 0 2 0 0 NXT	n0200-00	

All right, what have we learned? That each of the 1024 RAM locations can hold a double-hex digit number, be it data or instruction. We can write into any location just by keying in the desired two-digit number, any location, that is, except for those in the PROM memory circuits. If you don't believe it, try writing data into one of the PROM locations.

The PROM locations occur between addresses 8000H and 83FFH. If you access one of those locations and then try to store data into in, you'll find the data is not accepted.

Enter:	See Displayed:	Write Contents:
DCM 8 0 0 0 NXT	n8000-31	
F F	n8000-FF	
NXT	n8001-00	
DCM 8 0 0 0 NXT	n8000-31	

If you did this correctly you should have found 31H in memory location 8000H. When you keyed in FFH that number appeared in the displays as though it now resides in location 8000H, yet when you went back to 8000H, the location still contained 31H. There are two lessons here. Obviously you cannot write data into the PROMs; they won't accept it. The second lesson is an example of one of the most overlooked subtleties of the ia7301. Learn it now and save countless headaches.

IDIOSYNCRASY: BECAUSE THE ia7301 GIVES YOU A CHANCE TO CORRECT THE DATA YOU ARE TRYING TO ENTER INTO THE MEMORY IN THE SAME FASHION THAT IT ALLOWS YOU TO CORRECT AN ADDRESS, YOU MUST TELL THE COMPUTER THAT YOU ARE SATISFIED WITH THE CONTENTS OF THE LOCATION BEFORE IT WILL WRITE THAT DATA INTO MEMORY. YOU INFORM THE COMPUTER OF YOUR SATISFACTION BY PRESSING NXT .

We can see the effect of this subtle but important operation at work when we tried to write data into the PROMs that contain the system monitor. When we keyed FFH into the computer, that data appeared in the displays just as though it now resided in location 8000H. Yet as we later saw, it never actually reached the memory, for when we reexamined the same location nothing had been changed. Why was the computer seemingly trying to pretend that the data had been written when it had not?

The process of writing into the memory allows the user to examine the data as he has keyed it up prior to the actual memory loading action. If he has made a mistake or simply changes his mind, he can correct the error simply by keying in the corrected data. The actual loading process does not actually occur until the user cues the computer with the NXT key. Let's try an example. Try writing FFH into location 0100H without using the NXT key.

Enter:	See Displayed:	Write Contents:
DCM 0 1 0 0 NXT	n0100-	____
F F	n0100-FF	____
DCM	n-------	
0 1 0 0 NXT	n0100-	____

Well, did FFH appear in the memory? No, because you never told the computer you were happy with the data before you went on to something else by pressing DCM again. The computer assumed you changed your mind about wanting it to load FFH into 0100H and simply dropped the FFH that you had keyed into the displays. Now try it the right way using NXT to cue the computer to complete the loading process.

Enter:	See Displayed:	Write Contents:
DCM 0 1 0 0 NXT	n0100-	____
F F	n0100-FF	____
NXT	n0101-	
DCM 0 1 0 0 NXT	n0100-FF	____

This time you should have been able to successfully write the FFH into the memory. As we have pointed out, the difference is in the proper use of the NXT key. When entering programs most entries take place in successive locations so that the process of pressing NXT to increment the address automatically completes the loading process of the preceding location. This holds until the last location is reached and you must remember to press NXT this time or the last location will not be loaded properly.

SKILL II. BEFORE GOING ON TO CHAPTER THREE. YOU SHOULD KNOW HOW TO LOAD DATA INTO THE MEMORY, BOTH INTO SINGLE LOCATIONS AND STRINGS OF DATA INTO A SET OF ADJACENT LOCATIONS. IF YOU DO NOT FEEL COMFORTABLE WITH THE DCM KEY YET, TURN TO THE BACK OF THIS BOOK TO SUPPLEMENTARY EXERCISE II FOR ADDITIONAL PRACTICE ON THIS SUBJECT.